

What is claimed is:

1. A computer program product for improving performance and resource utilization of software applications that interact with a back-end data source to retrieve information stored therein, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for storing one or more objects in a cache for responding to read requests against the objects, wherein (1) a set of input properties and values thereof is stored with or associated with each stored object and (2) refresh logic specifying how to refresh each of the stored objects is stored with or associated with the stored object or a group of stored objects;

computer-readable program code means for specifying a refresh policy that corresponds to each stored object or to each group of stored objects;

computer-readable program code means for receiving read requests against one or more of the objects;

computer-readable program code means for responding to the read requests using the stored objects;

computer-readable program code means for scheduling a refresh of a selected stored object by queuing the selected stored object or a reference thereto as a queued refresh request on a refresh queue; and

computer-readable program code means for refreshing the selected stored object, when triggered according to the corresponding refresh policy, by executing the refresh logic stored with

21 or associated with the queued refresh request.

1 2. The computer program product according to Claim 1, wherein a separate refresh queue is
2 created for each of one or more back-end data sources to be accessed during operation of the
3 computer-readable program code means for refreshing.

1 3. The computer program product according to Claim 1, wherein the refresh policy
2 comprises information about an associated object which is used for responding to update requests.

1 4. The computer program product according to Claim 1, wherein the refresh policy
2 comprises reaching a particular time of day.

1 5. The computer program product according to Claim 1, wherein the refresh policy
2 comprises reaching an elapsed time since a prior refresh.

1 6. The computer program product according to Claim 1, further comprising:
2 computer-readable program code means for connecting to the back-end data source prior
3 to operation of the computer-readable program code means for refreshing; and
4 computer-readable program code means for disconnecting from the back-end data source
5 after operation of the computer-readable program code means for refreshing.

7. A computer program product for improving performance and resource utilization of

2 software applications that interact with a back-end data source to update information stored
3 therein, the computer program product embodied on one or more computer-readable media and
4 comprising:

5 computer-readable program code means for storing one or more objects in a cache for
6 responding to update requests against the objects, wherein (1) a set of input properties is stored
7 with or associated with each stored object and (2) update logic specifying how to update each of
8 the stored objects is stored with or associated with the stored object or a group of stored objects;

9 computer-readable program code means for receiving update requests against one or more
10 of the objects;

11 computer-readable program code means for determining an update mode to use for a
12 selected update request, responsive to the computer-readable program code means for receiving;

13 computer-readable program code means for immediately processing the selected update
14 request if the determined update mode is not a delayed update mode; and

15 computer-readable program code means for delaying processing of the selected update
16 request otherwise.

1 8. The computer program product according to Claim 7, wherein the computer-readable
2 program code means for delaying processing further comprises:

3 computer-readable program code means for queuing the selected update request, along
4 with the input properties and values thereof which are to be used for performing the selected
5 update request, as a queued update request on an update queue;

6 computer-readable program code means for detecting a triggering event for performing

7 the delayed processing of the queued update requests; and

8 computer-readable program code means for performing, responsive to the computer-
9 readable program code means for detecting, the queued update requests.

1 9. The computer program product according to Claim 8, wherein the computer-readable
2 program code means for performing further comprises:

3 computer-readable program code means for setting the input properties of a selected
4 object against which the queued update request is to be performed using the queued input
5 property values; and

6 computer-readable program code means for executing the update logic stored with or
7 associated with the selected object.

1 10. The computer program product according to Claim 8, wherein the triggering event
2 comprises reaching a particular count of queued update requests for a selected object.

1 11. The computer program product according to Claim 8, wherein the triggering event
2 comprises reaching a particular time of day.

1 12. The computer program product according to Claim 8, wherein the update policy
2 comprises information about an associated object which is used for responding to read requests.

1 13. The computer program product according to Claim 8, wherein a separate update queue is

2 created for each of one or more back-end data sources to be accessed during operation of the
3 computer-readable program code means for performing.

1 14. The computer program product according to Claim 7, wherein the computer-readable
2 program code means for determining further comprises computer-readable program code means
3 for selecting the delayed update mode based upon a time of day when the selected update request
4 is received.

1 15. The computer program product according to Claim 7, wherein the computer-readable
2 program code means for determining further comprises computer-readable program code means
3 for selecting the delayed update mode based upon a classification of a user making the selected
4 update request.

1 16. The computer program product according to Claim 8, further comprising:
2 computer-readable program code means for connecting to the back-end data source prior
3 to operation of the computer-readable program code means for performing; and
4 computer-readable program code means for disconnecting from the back-end data source
5 after operation of the computer-readable program code means for performing.

1 17. A system for improving performance and resource utilization of software applications that
2 interact with a back-end data source to retrieve information stored therein, comprising:
3 means for storing one or more objects in a cache for responding to read requests against

4 the objects, wherein (1) a set of input properties and values thereof is stored with or associated
5 with each stored object and (2) refresh logic specifying how to refresh each of the stored objects
6 is stored with or associated with the stored object or a group of stored objects;

7 means for specifying a refresh policy that corresponds to each stored object or to each
8 group of stored objects;

9 means for receiving read requests against one or more of the objects;

10 means for responding to the read requests using the stored objects;

11 means for scheduling a refresh of a selected stored object by queuing the selected stored
12 object or a reference thereto as a queued refresh request on a refresh queue; and

13 means for refreshing the selected stored object, when triggered according to the
14 corresponding refresh policy, by executing the refresh logic stored with or associated with the
15 queued refresh request.

1 18. The system according to Claim 17, wherein a separate refresh queue is created for each of
2 one or more back-end data sources to be accessed during operation of the means for refreshing.

1 19. The system according to Claim 17, wherein the refresh policy comprises information about
2 an associated object which is used for responding to update requests.

1 20. The system according to Claim 17, wherein the refresh policy comprises reaching a
2 particular time of day.

1 21. The system according to Claim 17, wherein the refresh policy comprises reaching an
2 elapsed time since a prior refresh.

1 22. The system according to Claim 17, further comprising:
2 means for connecting to the back-end data source prior to operation of the means for
3 refreshing; and
4 means for disconnecting from the back-end data source after operation of the means for
5 refreshing.

23. A system for improving performance and resource utilization of software applications that
2 interact with a back-end data source to update information stored therein, comprising:
3 means for storing one or more objects in a cache for responding to update requests against
4 the objects, wherein (1) a set of input properties is stored with or associated with each stored
5 object and (2) update logic specifying how to update each of the stored objects is stored with or
6 associated with the stored object or a group of stored objects;
7 means for receiving update requests against one or more of the objects;
8 means for determining an update mode to use for a selected update request, responsive to
9 the means for receiving;
10 means for immediately processing the selected update request if the determined update
11 mode is not a delayed update mode; and
12 means for delaying processing of the selected update request otherwise.

1 24. The system according to Claim 23, wherein the means for delaying processing further
2 comprises:

3 means for queuing the selected update request, along with the input properties and values
4 thereof which are to be used for performing the selected update request, as a queued update
5 request on an update queue;

6 means for detecting a triggering event for performing the delayed processing of the
7 queued update requests; and

8 means for performing, responsive to the means for detecting, the queued update requests.

1 25. The system according to Claim 24, wherein the means for performing further comprises:

2 means for setting the input properties of a selected object against which the queued update
3 request is to be performed using the queued input property values; and

4 means for executing the update logic stored with or associated with the selected object.

1 26. The system according to Claim 24, wherein the triggering event comprises reaching a
2 particular count of queued update requests for a selected object.

1 27. The system according to Claim 24, wherein the triggering event comprises reaching a
2 particular time of day.

1 28. The system according to Claim 24, wherein the update policy comprises information about
2 an associated object which is used for responding to read requests.

1 29. The system according to Claim 24, wherein a separate update queue is created for each of
2 one or more back-end data sources to be accessed during operation of the means for performing.

1 30. The system according to Claim 23, wherein the means for determining further comprises
2 means for selecting the delayed update mode based upon a time of day when the selected update
3 request is received.

1 31. The system according to Claim 23, wherein the means for determining further comprises
2 means for selecting the delayed update mode based upon a classification of a user making the
3 selected update request.

1 32. The system according to Claim 24, further comprising:
2 means for connecting to the back-end data source prior to operation of the means for
3 performing; and
4 means for disconnecting from the back-end data source after operation of the means for
5 performing.

1 33. A method for improving performance and resource utilization of software applications that
2 interact with a back-end data source to retrieve information stored therein, comprising the steps
3 of:
4 storing one or more objects in a cache for responding to read requests against the objects,

5 wherein (1) a set of input properties and values thereof is stored with or associated with each
6 stored object and (2) refresh logic specifying how to refresh each of the stored objects is stored
7 with or associated with the stored object or a group of stored objects;

8 specifying a refresh policy that corresponds to each stored object or to each group of
9 stored objects;

10 receiving read requests against one or more of the objects;

11 responding to the read requests using the stored objects;

12 scheduling a refresh of a selected stored object by queuing the selected stored object or a
13 reference thereto as a queued refresh request on a refresh queue; and

14 refreshing the selected stored object, when triggered according to the corresponding
15 refresh policy, by executing the refresh logic stored with or associated with the queued refresh
16 request.

1 34. The method according to Claim 33, wherein a separate refresh queue is created for each of
2 one or more back-end data sources to be accessed during operation of the refreshing step.

1 35. The method according to Claim 33, wherein the refresh policy comprises information
2 about an associated object which is used for responding to update requests.

1 36. The method according to Claim 33, wherein the refresh policy comprises reaching a
2 particular time of day.

1 37. The method according to Claim 33, wherein the refresh policy comprises reaching an
2 elapsed time since a prior refresh.

1 38. The method according to Claim 38, further comprising the steps of:
2 connecting to the back-end data source prior to operation of the refreshing step; and
3 disconnecting from the back-end data source after operation of the refreshing step.

39. A method for improving performance and resource utilization of software applications that
interact with a back-end data source to update information stored therein, comprising the steps of:
storing one or more objects in a cache for responding to update requests against the
objects, wherein (1) a set of input properties is stored with or associated with each stored object
and (2) update logic specifying how to update each of the stored objects is stored with or
associated with the stored object or a group of stored objects;
receiving update requests against one or more of the objects;
determining an update mode to use for a selected update request, responsive to the
receiving step;
immediately processing the selected update request if the determined update mode is not a
delayed update mode; and
delaying processing of the selected update request otherwise.

1 40. The method according to Claim 39, wherein the step of delaying processing further
2 comprises the steps of:

3 queuing the selected update request, along with the input properties and values thereof
4 which are to be used for performing the selected update request, as a queued update request on an
5 update queue;

6 detecting a triggering event for performing the delayed processing of the queued update
7 requests; and

8 performing, responsive to the detecting step, the queued update requests.

1 41. The method according to Claim 40, wherein the performing step further comprises the
2 steps of:

3 setting the input properties of a selected object against which the queued update request is
4 to be performed using the queued input property values; and

5 executing the update logic stored with or associated with the selected object.

1 42. The method according to Claim 40, wherein the triggering event comprises reaching a
2 particular count of queued update requests for a selected object.

1 43. The method according to Claim 40, wherein the triggering event comprises reaching a
2 particular time of day.

1 44. The method according to Claim 40, wherein the update policy comprises information
2 about an associated object which is used for responding to read requests.

1 45. The method according to Claim 40, wherein a separate update queue is created for each of
2 one or more back-end data sources to be accessed during operation of the step of performing.

1 46. The method according to Claim 39, wherein the determining step further comprises the
2 step of selecting the delayed update mode based upon a time of day when the selected update
3 request is received.

1 47. The method according to Claim 39, wherein the determining step further comprises the
2 step of selecting the delayed update mode based upon a classification of a user making the
3 selected update request.

1 48. The method according to Claim 40, further comprising the steps of:
2 connecting to the back-end data source prior to operation of the performing step; and
3 disconnecting from the back-end data source after operation of the performing step.